



ugr | Universidad
de **Granada**

PDIH

PERIFÉRICOS Y DISPOSITIVOS DE INTERFAZ HUMANA.

Práctica 3: Experimentación con Arduino.

Autora: Cristina María Crespo Arco

Correo: cmcrespo@correo.ugr.es

Autor: Andrés Piqueras Brück

Correo: andrespquieras@correo.ugr.es

Profesor: Pedro A. Castillo Valdivieso



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

Curso 2021 - 2022

Índice

1. Primer ejercicio.	2
1.1. Descripción del ejercicio	2
1.2. Componentes eléctricos.	2
1.3. Código.	2
1.4. Esquema de conexiones eléctricas.	3
1.4.1. Vista de Circuito.	3
1.4.2. Vista esquemática.	3
1.5. Funcionamiento del proyecto.	4
2. Segundo ejercicio: Versión interruptor.	5
2.1. Descripción del ejercicio	5
2.2. Componentes eléctricos.	5
2.3. Código.	5
2.4. Esquema de conexiones eléctricas.	6
2.4.1. Vista de Circuito.	6
2.4.2. Vista esquemática.	6
2.5. Funcionamiento del proyecto.	7
3. Segundo ejercicio: Versión pulsador.	8
3.1. Descripción del ejercicio	8
3.2. Componentes eléctricos.	8
3.3. Código.	8
3.4. Esquema de conexiones eléctricas.	9
3.4.1. Vista de Circuito.	9
3.4.2. Vista esquemática.	9
3.5. Funcionamiento del proyecto.	10
4. Tercer ejercicio.	12
4.1. Descripción del ejercicio	12
4.2. Componentes eléctricos.	12
4.3. Código.	12
4.4. Esquema de conexiones eléctricas.	13
4.4.1. Vista de Circuito.	13
4.4.2. Vista esquemática.	13
4.5. Funcionamiento del proyecto.	14
5. Cuarto ejercicio.	15
5.1. Descripción del ejercicio	15
5.2. Componentes eléctricos.	15
5.3. Código.	15
5.4. Esquema de conexiones eléctricas.	16
5.4.1. Vista de Circuito.	16
5.4.2. Vista esquemática.	16
5.5. Funcionamiento del proyecto.	17

1. Primer ejercicio.

1.1. Descripción del ejercicio

Implementar el programa de parpadeo de LED, ampliándolo para que encienda y apague alternativamente tres LEDs (uno rojo, otro amarillo y otro verde), conectados a las salidas digitales 11, 12 y 13 del Arduino, a un intervalo de 1.5 segundos. Crear el esquema con Fritzing y cargar el programa en Arduino para comprobar que funciona correctamente.

1.2. Componentes eléctricos.

- 1 Arduino Uno R3
- 3 Resistencias 220 Ω
- 1 LED Verde
- 1 LED Amarillo
- 1 LED Rojo
- 7 cables macho-macho

1.3. Código.

```
// C++ code
void setup(){
    // put your setup code here, to run once:
    pinMode(11, OUTPUT);      //pin de salida de la LED roja
    pinMode(12, OUTPUT);      //pin de salida de la LED amarilla
    pinMode(13, OUTPUT);      //pin de salida de la LED verde
}

void loop(){
    // put your main code here, to run repeatedly:
    digitalWrite(13, LOW);    //Apagar LED verde
    digitalWrite(11, HIGH);   //Encender LED roja
    delay(1500);             // Esperar durante 1,5 segundos
    digitalWrite(11, LOW);    //Apagar LED roja
    digitalWrite(12, HIGH);   //Encender LED amarilla
    delay(1500);             // Esperar durante 1,5 segundos
    digitalWrite(12, LOW);    //Apagar LED amarilla
    digitalWrite(13, HIGH);   //Encender LED verde
    delay(1500);             // Esperar durante 1,5 segundos
}
```

1.4. Esquema de conexiones eléctricas.

1.4.1. Vista de Circuito.

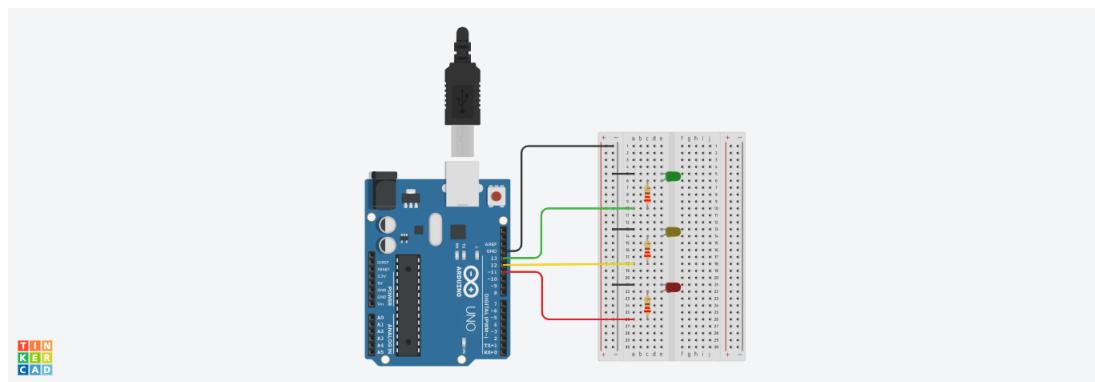


Figura 1: Circuito.

1.4.2. Vista esquemática.

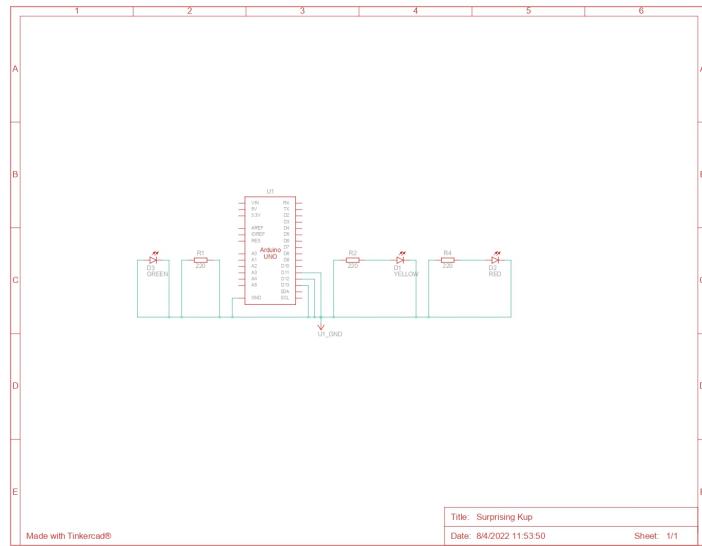


Figura 2: Esquema.

1.5. Funcionamiento del proyecto.

- ### ■ Circuito.

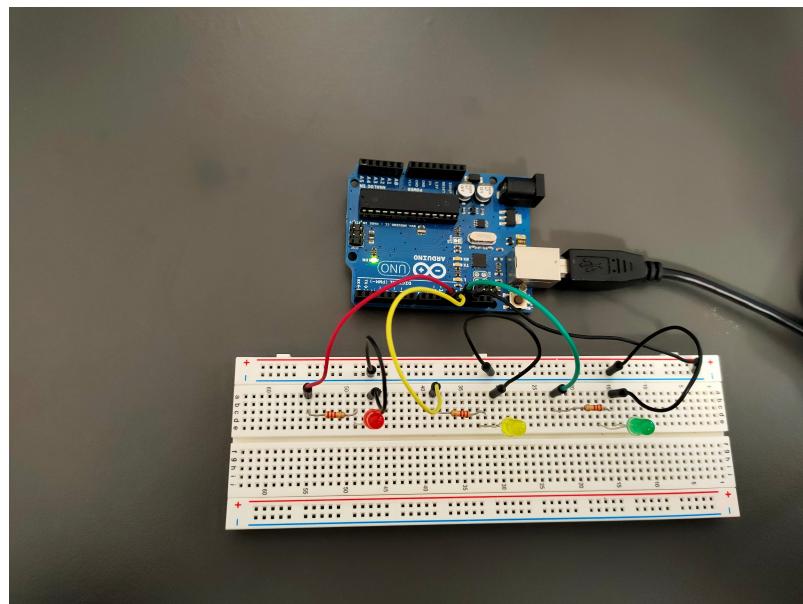


Figura 3: Arduino.

- #### ■ Vídeos.

https://github.com/andres003/pdih/blob/main/P3/Ejercicio1/ejerV1_1.mp4

https://github.com/andres003/pdih/blob/main/P3/Ejercicio1/ejerV1_2.mp4

2. Segundo ejercicio: Versión interruptor.

2.1. Descripción del ejercicio

Partir del programa de parpadeo de LEDs anterior y ampliarlo con las modificaciones necesarias para que se encienda el LED rojo solo cuando se pulse un interruptor conectado a la entrada digital 7, y en ese momento se apaguen los LEDs amarillo y verde.

2.2. Componentes eléctricos.

- 1 Arduino Uno R3
- 4 Resistencias 220 Ω
- 1 LED Verde
- 1 LED Amarillo
- 1 LED Rojo
- 1 Interruptor deslizante
- 11 cables macho-macho

2.3. Código.

```
// C++ code
void setup(){
    // put your setup code here, to run once:
    pinMode(7, INPUT);           //pin de entrada del INTERRUPTOR
    pinMode(11, OUTPUT);         //pin de salida de la LED roja
    pinMode(12, OUTPUT);         //pin de salida de la LED amarilla
    pinMode(13, OUTPUT);         //pin de salida de la LED verde
}
void loop(){
    // put your main code here, to run repeatedly:
    //Comprueba si el interruptor está encendido
    if ((digitalRead(7) == true)) {
        digitalWrite(11, HIGH);      //Encender LED roja
        digitalWrite(12, LOW);       //Apagar LED amarilla
        digitalWrite(13, LOW);       //Apagar LED verde
    }
    //Comprueba si el interruptor está apagado
    else if ((digitalRead(7) == false)) {
        digitalWrite(11, LOW);      //Apagar LED roja
        digitalWrite(12, HIGH);      //Encender LED amarilla
        digitalWrite(13, HIGH);      //Encender LED verde
    }
}
```

2.4. Esquema de conexiones eléctricas.

2.4.1. Vista de Circuito.

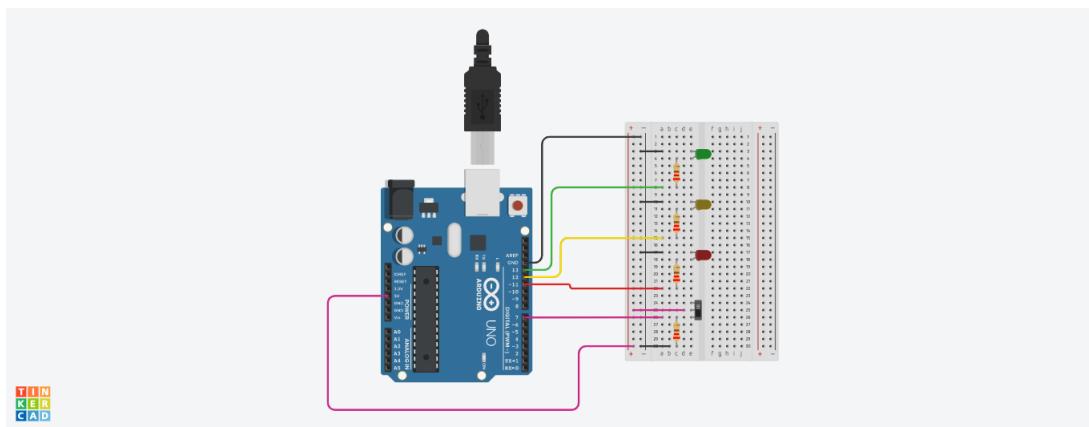


Figura 4: Circuito.

2.4.2. Vista esquemática.

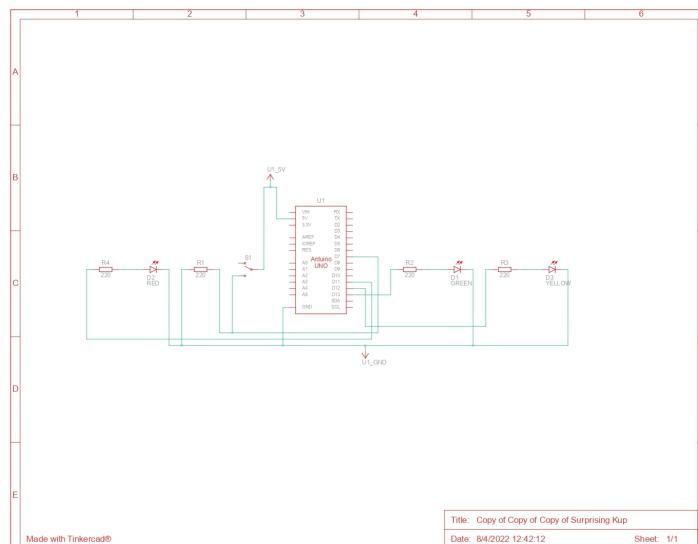


Figura 5: Esquema.

2.5. Funcionamiento del proyecto.

- Circuito.

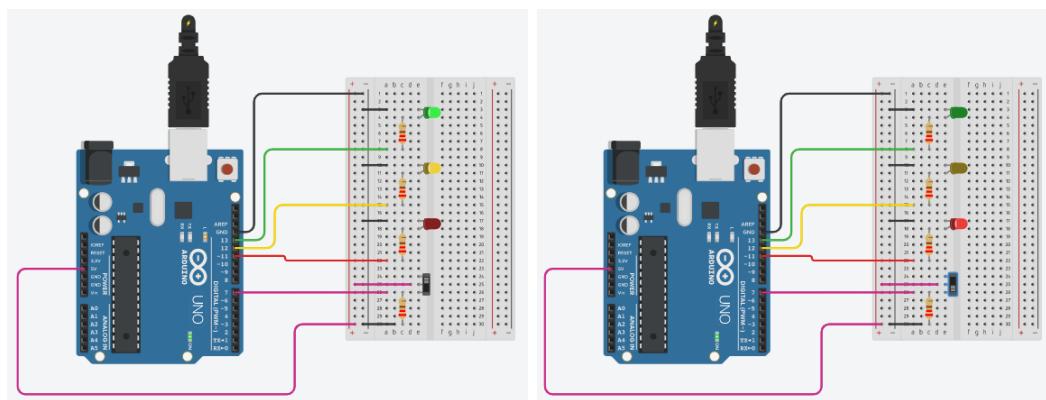


Figura 6: Interruptor apagado - Interruptor encendido.

- Vídeo.

https://github.com/andres003/pdih/blob/main/P3/Ejercicio2/ejerV2_1.mp4

3. Segundo ejercicio: Versión pulsador.

3.1. Descripción del ejercicio

Partir del programa de parpadeo de LEDs anterior y ampliarlo con las modificaciones necesarias para que se encienda el LED rojo solo cuando se pulse un interruptor conectado a la entrada digital 7, y en ese momento se apaguen los LEDs amarillo y verde.

3.2. Componentes eléctricos.

- 1 Arduino Uno R3
- 4 Resistencias 220 Ω
- 1 LED Verde
- 1 LED Amarillo
- 1 LED Rojo
- 1 Pulsador
- 11 cables macho-macho

3.3. Código.

```
// C++ code
void setup(){
    // put your setup code here, to run once:
    pinMode(7, INPUT);           //pin de entrada del INTERRUPTOR
    pinMode(11, OUTPUT);         //pin de salida de la LED roja
    pinMode(12, OUTPUT);         //pin de salida de la LED amarilla
    pinMode(13, OUTPUT);         //pin de salida de la LED verde
}
void loop(){
    // put your main code here, to run repeatedly:
    //Comprueba si el pulsador está pulsado
    if ((digitalRead(7) == true)) {
        digitalWrite(11, HIGH);      //Encender LED roja
        digitalWrite(12, LOW);       //Apagar LED amarilla
        digitalWrite(13, LOW);       //Apagar LED verde
    }
    //Comprueba si el pulsador no está pulsado
    else if ((digitalRead(7) == false)) {
        digitalWrite(11, LOW);      //Apagar LED roja
        digitalWrite(12, HIGH);     //Encender LED amarilla
        digitalWrite(13, HIGH);     //Encender LED verde
    }
}
```

3.4. Esquema de conexiones eléctricas.

3.4.1. Vista de Circuito.

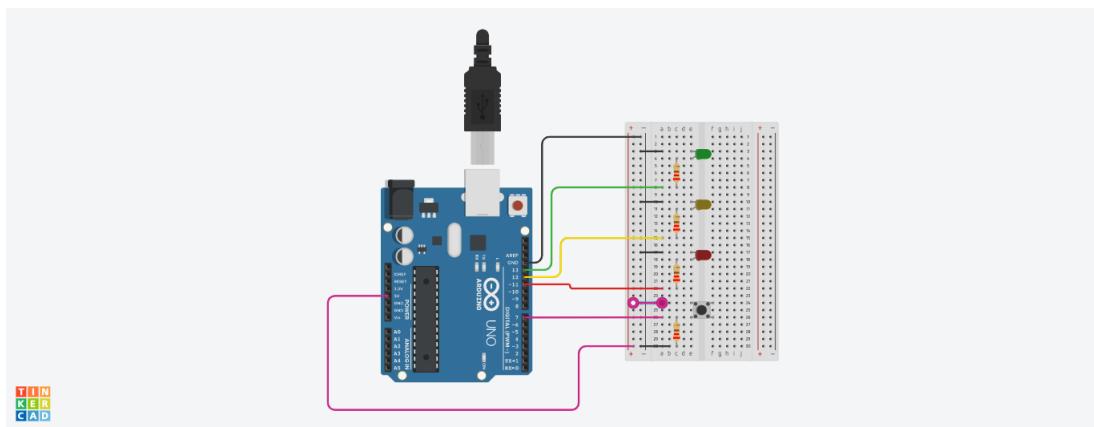


Figura 7: Circuito.

3.4.2. Vista esquemática.

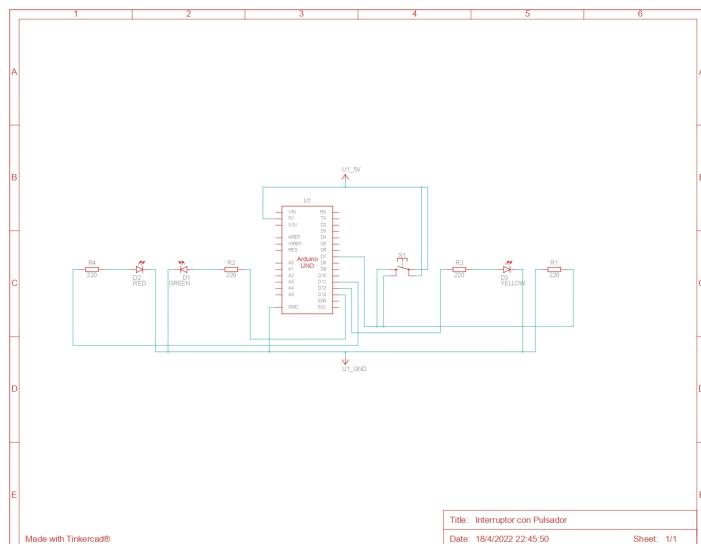


Figura 8: Esquema.

3.5. Funcionamiento del proyecto.

- Circuito.

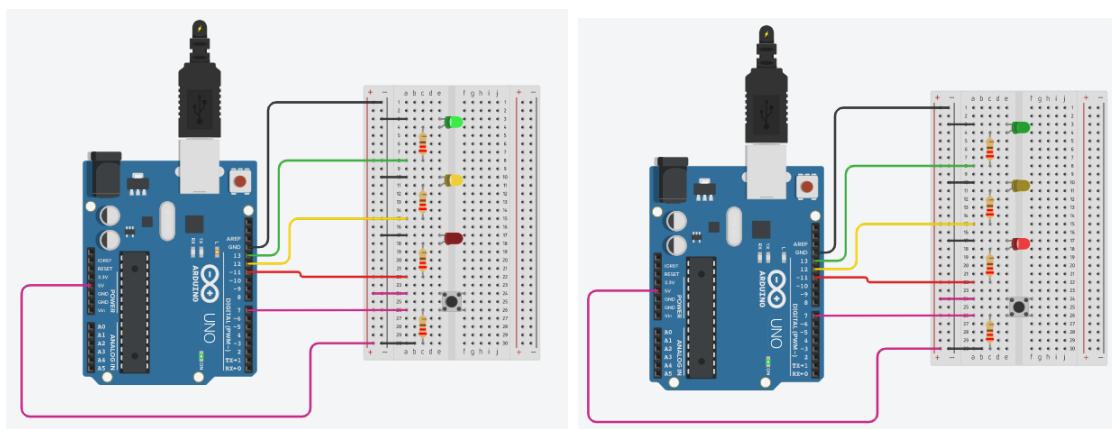


Figura 9: Interruptor apagado - Interruptor encendido.

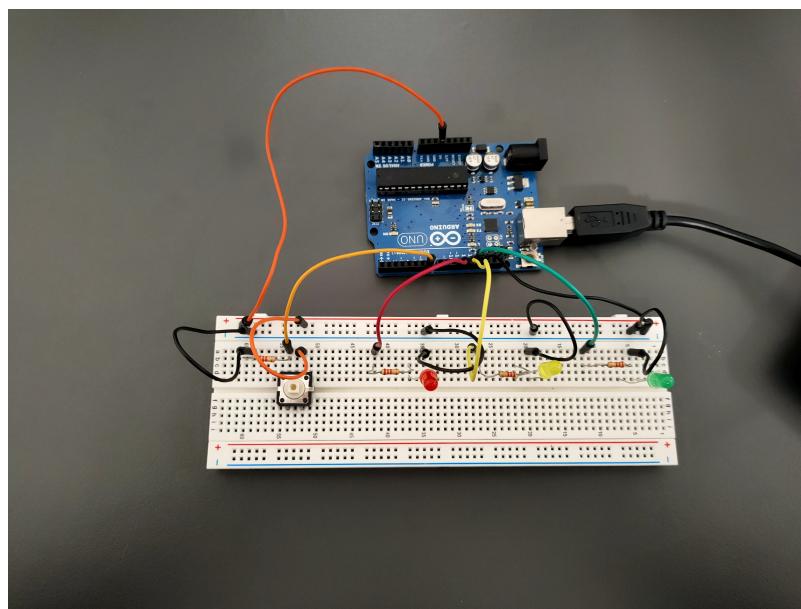


Figura 10: Interruptor apagado - Interruptor encendido.

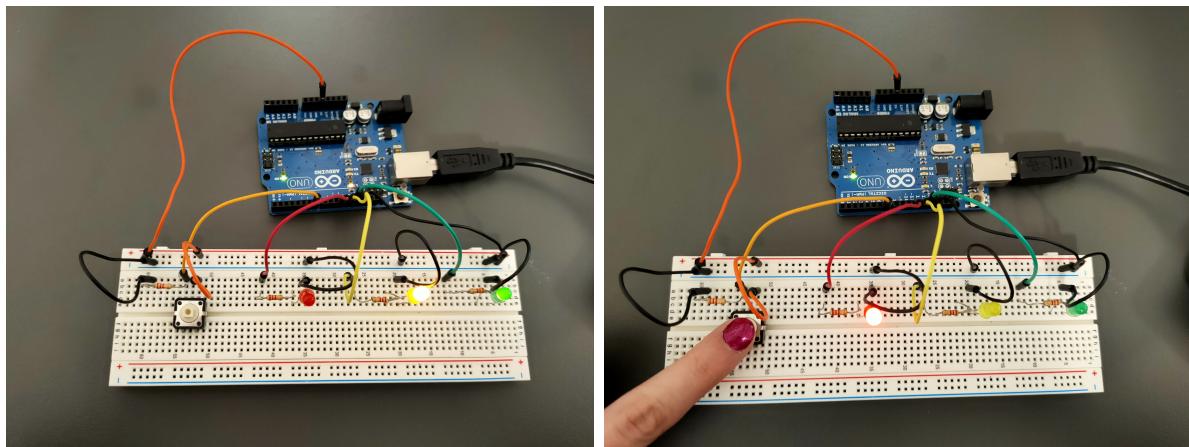


Figura 11: Interruptor apagado - Interruptor encendido.

- Vídeos.

https://github.com/andres003/pdih/blob/main/P3/Ejercicio2/ejerV2_11.mp4
https://github.com/andres003/pdih/blob/main/P3/Ejercicio2/ejerV2_12.mp4

4. Tercer ejercicio.

4.1. Descripción del ejercicio

Secuencia de LEDs, encendiendo y apagando 4 LEDs secuencialmente, de forma similar a las luces de “El coche fantástico”.

4.2. Componentes eléctricos.

- 1 Arduino Uno R3
- 4 Resistencias 220 Ω
- 4 LED Rojo
- 9 cables macho-macho

4.3. Código.

```
// C++ code
void setup(){
    // put your setup code here, to run once:
    pinMode(10, OUTPUT);      //pin de salida de una LED
    pinMode(11, OUTPUT);      //pin de salida de una LED
    pinMode(12, OUTPUT);      //pin de salida de una LED
    pinMode(13, OUTPUT);      //pin de salida de una LED
}
void loop(){
    // put your main code here, to run repeatedly:
    digitalWrite(11, LOW);    //Apagar LED
    digitalWrite(10, HIGH);    //Encender LED
    delay(200);              // Esperar durante 0,2 segundos
    digitalWrite(10, LOW);    //Apagar LED
    digitalWrite(11, HIGH);    //Encender LED
    delay(200);              // Esperar durante 0,2 segundos
    digitalWrite(11, LOW);    //Apagar LED
    digitalWrite(12, HIGH);    //Encender LED
    delay(200);              // Esperar durante 0,2 segundos
    digitalWrite(12, LOW);    //Apagar LED
    digitalWrite(13, HIGH);    //Encender LED
    delay(200);              // Esperar durante 0,2 segundos
    digitalWrite(13, LOW);    //Apagar LED
    digitalWrite(12, HIGH);    //Encender LED
    delay(200);              // Esperar durante 0,2 segundos
    digitalWrite(12, LOW);    //Apagar LED
    digitalWrite(11, HIGH);    //Encender LED
    delay(200);              // Esperar durante 0,2 segundos
}
```

4.4. Esquema de conexiones eléctricas.

4.4.1. Vista de Circuito.

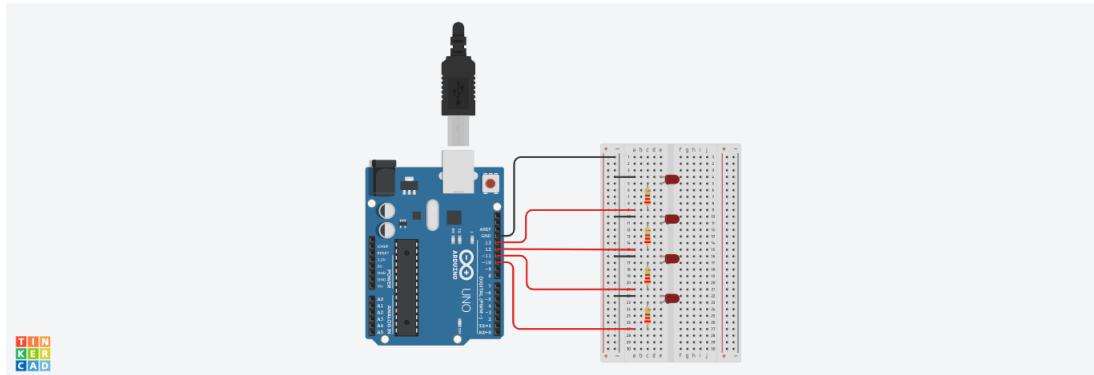


Figura 12: Circuito.

4.4.2. Vista esquemática.

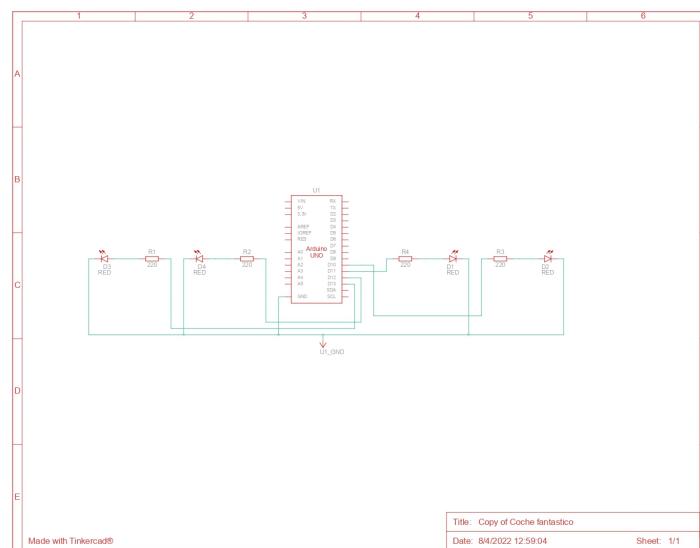


Figura 13: Esquema.

4.5. Funcionamiento del proyecto.

- Circuito.

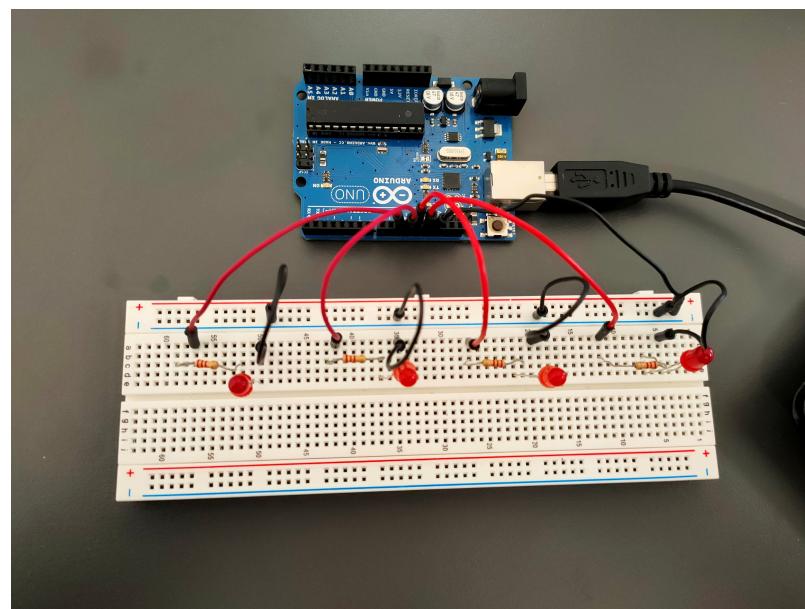


Figura 14: Arduino.

- Vídeos.

https://github.com/andres003/pdih/blob/main/P3/Ejercicio3/ejerV3_1.mp4
https://github.com/andres003/pdih/blob/main/P3/Ejercicio3/ejerV3_2.mp4

5. Cuarto ejercicio.

5.1. Descripción del ejercicio

Alarma por detección de presencia.

5.2. Componentes eléctricos.

- 1 Arduino Uno R3
- 2 Resistencias 220 Ω
- 1 LED Rojo
- 1 Sensor PIR
- 7 cables macho-macho
- 3 cables macho-hembra

5.3. Código.

```
// C++ code
void setup(){
    // put your setup code here, to run once:
    pinMode(13, OUTPUT);           //pin de salida de la LED roja
    pinMode(7, INPUT);            //pin de entrada del SENSOR PIR
}

void loop(){
    // put your main code here, to run repeatedly:
    //Comprueba si hay movimiento cerca del sensor PIR
    if ((digitalRead(7) == HIGH)) {
        digitalWrite(13, HIGH);      //Encender LED roja
    } else {
        digitalWrite(13, LOW);       //Apagar LED roja
    }
}
```

5.4. Esquema de conexiones eléctricas.

5.4.1. Vista de Circuito.

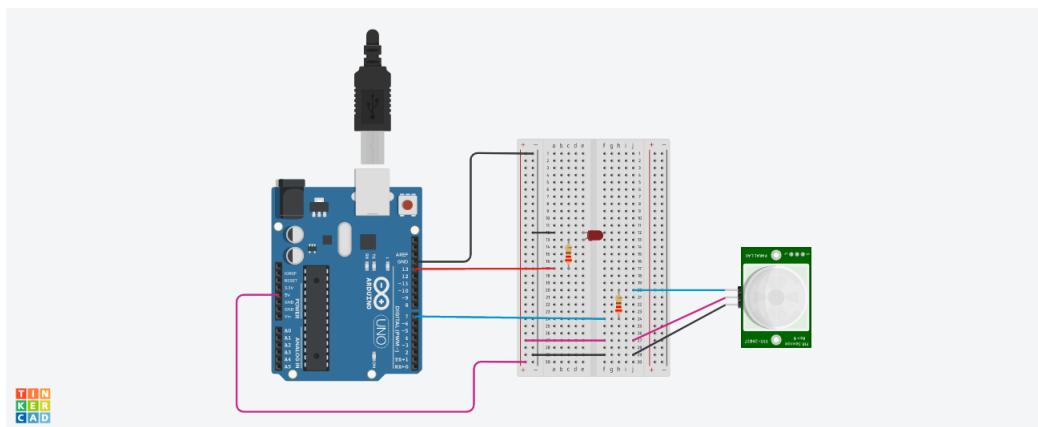


Figura 15: Circuito.

5.4.2. Vista esquemática.

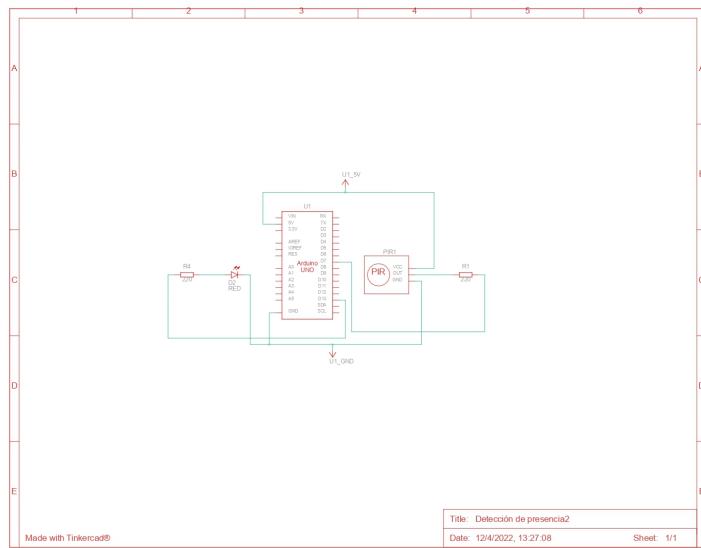


Figura 16: Esquema.

5.5. Funcionamiento del proyecto.

- Circuito.

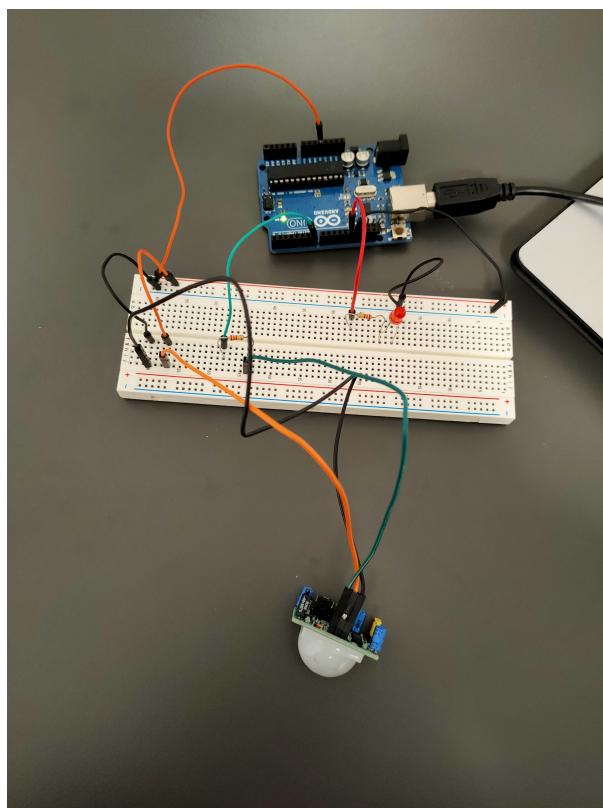


Figura 17: Arduino.

- Vídeos.

https://github.com/andres003/pdih/blob/main/P3/Ejercicio4/ejerV4_1.mp4

https://github.com/andres003/pdih/blob/main/P3/Ejercicio4/ejerV4_2.mp4